

Convex Hull-Based Multi-objective Genetic Programming for Maximizing ROC Performance

Pu Wang^a, Michael Emmerich^b, Rui Li^b, Ke Tang^a, Thomas Bäck^b, Xin Yao^c

^a*Nature Inspired Computation and Applications Laboratory (NICAL)
School of Computer Science and Technology*

University of Science and Technology of China (USTC), Hefei, Anhui 230027, China.

^b*Leiden Institute for Advanced Computer Science (LIACS),
Leiden University, 2333 CA Leiden, Netherlands.*

^c*Center of Excellence for Research in Computational Intelligence and Applications (CERCIA)
School of Computer Science,
The University of Birmingham Edgbaston,
Birmingham B15 2TT, U.K.*

Abstract

Receiver operating characteristic (ROC) is usually used to analyse the performance of classifiers in data mining. An important ROC analysis topic is ROC convex hull (ROCCH), which is the least convex majorant (LCM) of the empirical ROC curve, and covers potential optima for the given set of classifiers. Generally, ROC performance maximization could be considered to maximize the ROCCH, which also means to maximize the true positive rate (*tpr*) and minimize the false positive rate (*fpr*) for each classifier in the ROC space. However, *tpr* and *fpr* are conflicting with each other in the ROCCH optimization process. Though ROCCH maximization problem seems like a multi-objective optimization problem (MOP), the special characters make it different from traditional MOP. In this work, we will discuss the difference between them and propose convex hull-based multi-objective genetic programming (CH-MOGP) to solve ROCCH maximization problems. Convex hull-based sort is an indicator based selection scheme that aims to maximize the area under convex hull, which serves as an unary indicator for the performance of a set of points. A selection procedure is described that can be efficiently implemented and follows similar design principles than classical hypervolume based optimization algorithms. It is hypothesized that by using a tailored indicator-based selection scheme CH-MOGP gets more efficient for ROC convex hull approximation than algorithms which compute all Pareto optimal points. To test our hypothesis we compare the new CH-MOGP to MOGP with classical selection schemes, including Non-dominated Sorting Genetic Algorithm-II (NSGA-II), Multi-objective Evolutionary Algorithms Based on Decomposition (MOEA/D) and Multi-objective Selection Based on Dominated Hypervolume (SMS-EMOA). Experimental results based on 22 well-known UCI data sets show that CH-MOGP outperforms significantly traditional EMOAs.

*Emails: wuyou308@mail.ustc.edu.cn, emmerich@liacs.nl

Keywords: Classification, ROC analysis, AUC, ROCCH, Genetic Programming, Evolutionary Multi-objective Algorithm, Memetic Algorithm, Decision tree, Hypervolume Indicator.

1. Introduction

Traditionally, a classification task is to assign items (instances) in a data-set to target categories (classes) based on classifier(s) learnt by training instances. In binary classification there are only two classes or categories and all instances in the data set will be assigned one of them. The target of a classification problem is trying to design classifiers which make error-free assignments.

The ROC graph is a technique for visualizing, organizing and selecting classifiers based on their performance [1]. A salient topic in ROC analysis is to generate ROC curves for varying discriminative thresholds over the output of the classifier [1], and ROC curves have been used widely in many areas. Actually, over the course of the past 40 years, ROC technique has been widely applied in many research and application areas, such as signal detection [2], medical decision making [3], diagnostic systems [4].

Though ROC curve works well in many cases, recently attention of the research is also drawn towards another perspective of ROC analysis, namely ROC convex hull (ROCCH). ROCCH pays more attention to the convex hull of a set of points (hard classifiers) obtained either from sever curves (i.e., soft classifiers) or itself (hard classifier). A classifier is potentially optimal, if and only if it is a component of ROCCH, in other words, ROCCH could provide better choices than a single ROC curve to specific environments. The significance of ROCCH in ROC analysis is that for test data sets with different skewed class distributions or misclassification costs, it is always possible to choose suitable classifiers by iso-performance lines¹ which is translated by operating conditions of classifiers and used to identify a portion of the ROCCH [5]. Consequently, ROCCH is emphasized in this paper and we will focus on searching a group of independently hard classifiers to maximize the ROCCH performance rather try to maximize the area under the ROC curve (AUC) of a single soft classifier.

Essentially, ROCCH is the collection of all potentially optimal classifiers in a given set of classifiers, so ROCCH maximization is to find a group of classifiers with their performance approximating the top and the left axes as near as possible in ROC space. However, ROCCH maximization is not an easy task, there are not many works focusing on how to maximize the ROCCH though it is a really important topic in classification problems. Generally, the exist works could be reviewed into two categories, ROC geometric analysis based machine learning methods and multi-objective optimization strategies based evolutionary computation methods for ROCCH maximization.

Fawcett et al. [6] employed C4.5 and Rule Learning (RL) systems to induce decision rules in ROC space and its advanced version PRIE was introduced in [7]. It was a straight way to analysis the geometrical properties to generate decision rules to

¹All classifiers corresponding to the points on one line have the same expected costs.

maximize the ROC performance. However, the procedure easily gets trapped in local optima.

The concavity problem in ROC analysis was researched by Flach et al. [8] who demonstrated how to detect and repair concavities in ROC curves. The basic idea of that work is that if a point in the concavity can be mirrored to a better point which could perform well beyond the original ROC curve. But it is not a general method to maximize the ROC performance.

ROCCER was introduced by Prati et al. in [9]. It was argued that ROCCER is less dependent on the previously induced rules compared with set covering algorithms to construct rule sets that have a convex hull in ROC space. However, it adopted an association rule learner to generate new rules to cover the instance space as full as possible. It is too easy to fall into overfitting, because it needs many rules to cover the space which is similar with a decision tree with a very high height.

The Neyman-Pearson lemma as the theoretical basis for finding the optimal combination of classifiers to maximize the ROCCH is given in [10]. In contrast to the similar technique in [8], it not only focuses on repairing but it also pays attention on improving if there was on concavity. For a given rule set, the method proposed by [10] can be efficient to combine these rules using *AND* and *OR* to get the optimum rule subset. However, as mentioned above, it misses schemes for generating new rules in the global rule set searching.

To maximize ROCCH is searching a group of classifiers to maximize the ROCCH performance ideally would yield classifiers that simultaneously minimize the *fpr* and maximize the *tpr*, i. e. that are located as much to the left and to the top of the ROC space as possible. However, it is very hard to optimize *fpr* and *tpr* simultaneously because they are conflicting targets. From this perspective, ROCCH maximization problem is similar to multi-objective optimization problem.

Zhao [11] proposed specific non-dominated relationship involved into multi-objective optimization framework to optimize *tpr* and $1-fpr$. However, it paid more attention on cost-sensitive classification and made many rules by information of costs of misclassification to rank the individuals in its multi-objective genetic programming. First, it is not a general method for ROCCH maximization because it only focused on cost-sensitive problem. Second, two data sets involved in experiments are too few to evaluation the proposed method.

Bhowan et al. searched the Pareto front to maximize the accuracy of each minority class with unbalanced data set [12], and they also employed multi-objective optimization techniques to evolve diverse ensembles using genetic programming to maximize the classification performance in [13].

Wang et al. investigated investigated some EMOAs such as NSGA-II [14], MOEA/D [15], SMS-EMOA [16] and Approximation-Guided Evolutionary Multi-objective Algorithm (AG-EMOA) [17]. These different evolutionary multi-objective optimization frameworks had been combined with genetic programming to maximize ROC performance [18].

However, ROCCH is different from Pareto front though it was reported they were similar to each other [19]. ROCCH is the collection of points which construct the convex hull of existing classifiers in ROC space, and Pareto front is the collection of points that is the first level sorted by dominance relationship. Though evolutionary multi-objective algorithms(EMOAs) have been successfully used into ROCCH max-

imization, these EMO techniques do not take into account a special characteristic of ROCCH. That is by mixing two classifiers we can take any two real classifiers to construct any virtual classifier with its performance as a point along the line connected by above two points [19]. Consequently, hard classifiers in concave parts of the Pareto front can always be replaced by classifier combinations that yield dominating points. The computational resources for the approximation of concave parts are thus better spent on the accurate approximation of only those parts of the Pareto front that are part of the convex hull.

In [20, 21, 22], convex hull concept of was employed into EMOAs to make the sort fast or maintain a well-distributed set of non-dominated solutions. These work are good to supply some ideas of convex hull based sort. In [23] and [24], convex hull-based ranking involved with evolutionary multi-objective optimization and fuzzy rule-based binary classifiers to maximize ROOCH in ROC space. However, the number of levels was pre-defined as three without explaining in first work and the second one was considered as bi-objectives optimization, which were accuracy of classification and complexity of classifier rules.

Moreover, instead of designing algorithms based on Pareto dominance compliant performance indicators, such as the hypervolume indicator as done in [16] and in [25], it seems more promising to directly target the algorithm towards the maximization of the area under the convex hull (AUCH).

In this paper, we utilize Genetic Programming (GP) combined multi-objective techniques to get the optimal ROCHH. Two strategies will be represented, the first is the convex hull-based without redundancy sort to make the population of GP into several levels such as non-dominated sort in NSGA-II, the second is using area-based contribution to select the survivors in the same level, actually we use $\mu + \mu$ selection strategy as [25]. We show that convex hull-based without redundancy sort plays a key role in multi-objective genetic programming (MOGP) for maximizing ROCCH performance and area-based contribution selection scheme also can improve the performance.

This paper is organized as follows: Section 2 will discuss the relationship between ROCCH optimization and traditional multi-objective optimization in detail. Convex hull-based multi-objective genetic programming (CH-MOGP) will be described in Section 3. Experiments are studied in Section 4 and shows the advantages of our new algorithm. Section 5 gives the conclusions and a discussion on the important aspects and the future perspectives of this work.

2. ROC Convex Hull and Multi-objective Optimization

2.1. What is ROCCH?

Basically, ROC analysis concerns the confusion matrix for the outputs of a classifier, in which we can analysis the performance by measuring different metrics such as accuracy, precise, specificity, sensitivity and some others. ROC graph (Left side of Fig. 1) is plotted upon Y axis and X axis respectively taken tpr and fpr , which are also defined from the confusion matrix. Each classifier can be mapped in the ROC graph by its performance. Essentially, ROCCH is the collection of all potentially optimal classifiers in a given set of classifiers(Right side of Fig. 1). Furthermore, a classifier is

potentially optimal if and only if it lies on the convex hull of the set of points in ROC space [1].

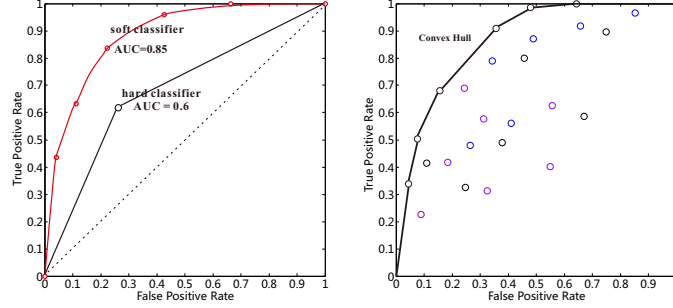


Figure 1: ROC graph and ROC Convex hull in ROC Space

2.2. ROCCH maximization problem and multi-objective optimization problem

The target of ROOCH maximization problem essentially aims at searching a group of solutions (classifiers) to approximate the upmost line and the leftmost line in ROC space as closely as possible. However, it is conflicting to minimize fpr and maximize tpr simultaneously because if the classifier labels more instances as positives, it will produce less negatives and vice versa. Generally speaking, ROCCH maximization is considered as a multi-objective optimization problem from this perspective and we can describe it as follows:

$$\begin{aligned} \text{maximize } F(x) &= (f_{tpr}(x), f_{1-fpr}(x)) \\ \text{subject to } &x \in \Omega \end{aligned} \quad (1)$$

In Eq. 1, x is a classifier and $F(x)$ is a vector function for fpr and tpr of the classifier. An important term in MOP is *dominance* which can be defined as: Let $u = (u_1, \dots, u_m)$, $v = (v_1, \dots, v_m)$ be two vectors, u is said to *dominate* v if $u_i \leq v_i$ for all $i = 1 \dots m$, and $u \neq v$, this is noted as $u \prec v$. If u and v can not dominate each other, we say that u and v are *nondominated*. The nondominated set is a set that each item does not dominate any another one. A point x^* is called *Pareto optimal* if there is no $x \in \Omega$ such that $F(x)$ dominates $F(x^*)$ [15, 26]. Pareto set (PS) is the collection of all Pareto optimal points. The Pareto front is the set of all the Pareto objective vectors $PF = \{F(x) | x \in PS\}$.

Most evolutionary multi-objective algorithms involves the pair-wise based dominance to describe the relationship of two solutions. However, we get a special character in ROCCH maximization in ROC space. Fig. 2 shows the convex hull part and Pareto front for all points. Obviously, convex hull is different from the Pareto front though they were argued that they are similar to each other [27]. For example, points a, b, c in Fig. 2 are non-dominated set in traditional multi-objective optimization problem, however, the classifier along the line connected by a and c would dominate b . That is the

special character in ROC maximization problem which makes ROCCH maximization is beyond traditional multi-objective optimization. However, we need to design some new techniques for searching a group of classifiers with maximum ROCCH.

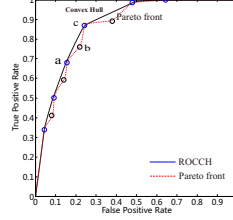


Figure 2: Pareto front and convex hull

2.3. Nondominated sort does harm to EMOAs in ROCCH maximization

The root reason for why we want to get the convex hull rather than Pareto front is that two classifiers will produce any classifiers with their ROC performance which is along the line connected by two point representing for the performance for previous two classifiers in ROC space [19]. As shown in left side of Fig. 3, classifiers with performance at point d and b can be used to construct any virtual classifier with its performance at e along the line connected by d and b . That is a special and important character in ROCCH maximization problem.

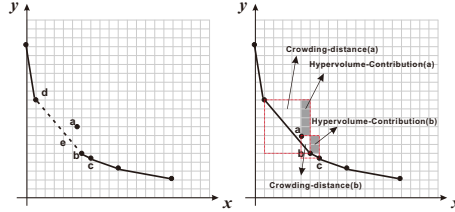


Figure 3: Nondominated sort keeps the individual which does nothing contribution to ROCCH

In the right side of Fig. 3, all the points are nondominated to each other and belong to the convex hull except for point a . However, if we take crowding-distance selection or hyper-volume contribution based selection to select one individual to be discarded from the population, obviously, point a will be selected to survive rather than point b though point a is not on the convex hull. Actually, there are two phenomenons we need pay attention to, one is the sort strategy and the other is the selection scheme. Besides, suitable sort strategy and selection scheme are should be considered in EMOAs for ROCCH not matter which classifier is involved.

2.4. The motivation and ideas for new multi-objective algorithms for ROCCH maximization

We need to think about how to use the special character of ROCCH to make multi-objective optimization techniques more efficient to solve the ROCCH maximization

problem. The main techniques for MOP is how to rank the population to select the solutions to survive in next generation. The mostly common rank approach includes two steps, one is sorting the population into several levels indicating the priority level, after that, a selection scheme is used to choose winners from solutions at the same level. In ROCCH maximization problem, firstly, convex hull-based idea will be considered into sorting strategy, however, because of the critical concept of convex hull, it would make the diversity decrease fast in the evolutionary process, so we design convex hull-based sorting without redundancy to sort the population. Another idea is to use area-base selection scheme because the target is to maximize the area under the convex hull instead of hypervolume or crowding-distance. Convex hull-based sorting without redundancy and area-based selection scheme will be described in detail in Section 3.

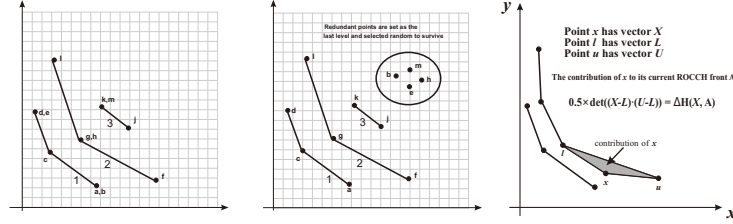


Figure 4: Convex hull-based sorting with and without redundancy, Area-based contribution to ROCCH

3. Convex Hull-based Multi-objective Genetic Programming (CH-MOGP)

In this section, we will describe our proposed convex hull-based multi-objective genetic programming to maximize ROCCH. Firstly, convex hull-based sorting without redundancy approach is used to rank the individuals in the union population into several levels which represent different priorities to survive as described in NSGA-II. Secondly, as the target is to maximize the area under the convex hull (AUCH) rather than the hypervolume mentioned in SMS-EMOA, and area-based indicator is designed to calculate the contribution of each individual to AUCH maximization. One major disadvantage of $(\mu + 1)$ selection strategy was employed in SMS-EMOA and AG-EMOA is that it needs to call fast-nondominated sorting μ times to select μ offsprings. In [25], an approximate scheme $(\mu + \mu)$ is proposed to make the selection process faster, and this idea has been adopted in CH-MOGP.

3.1. Convex hull-based without redundancy sorting

First of all, we introduce convex hull-based without redundancy sorting in this subsection. The main idea is to keep the diversity of the population by force, that means, each redundant solution will be put into an archive to be randomly selected to survive into the next generation if there are not enough non-redundant solutions to fill the whole population. Non-redundant solutions with not good performance have a chance to be kept by discarding the redundant solutions with good performance to make high diversity, and this could avoid that the solutions at the convex hull being copied a lot at the selection phase in evolutionary multi-objective optimization. As described in Alg. 3,

Algorithm 1 *Convex hull-based-sorting-without-redundancy* (Q, r)

Require: $Q \neq \emptyset$

- 1: Q is a solution set
- 2: r is the reference point

Ensure: ch-based-sorting-without-redundancy

- 3: $i = 0$
 - 4: **while** $Q \neq \emptyset$ **do**
 - 5: $T = Q \cup \{r\}$
 - 6: $\mathbf{F}_i = \text{Jarvis-Algorithm}(T)$ [28]
 - 7: $\mathbf{F}_i = \text{Elimination}(\mathbf{F}_i)$ // Some points in \mathbf{F}_i are not interesting and removed
 - 8: $Q = Q - \mathbf{F}_i$
 - 9: $i = i + 1$
 - 10: **end while**
-

the population will be split into redundant part and the other part which is sorted by convex hull-based sorting into several levels and the redundant part is taken as the last level which is the candidates by random selecting.

In Fig. 4, the first and second graphs gives the illustration for convex hull-based sorting with and without redundancy. All the redundant individuals will be discarded into the last level and selected random to the next generation if it is necessary.

Algorithm 2 *DeltaArea* (Q)

Require: $Q \neq \emptyset$

- 1: Q is a solution set

Ensure: DeltaArea

- 2: $m = \text{sizeof}(Q)$
 - 3: \mathbf{E} is performance of Q
 - 4: $\text{DeltaH}_1, \dots, \text{DeltaH}_m \leftarrow 0$
 - 5: **if** $m < 3$ **then**
 - 6: Set $\text{DeltaH}_1, \dots, \text{DeltaH}_m \leftarrow \infty$
 - 7: **else**
 - 8: Set $\text{DeltaH}_1, \text{DeltaH}_m \leftarrow \infty$
 - 9: **for** $2 \leq i \leq \text{sizeof}(Q) - 1$ **do**
 - 10: $\text{DeltaH}_i = 0.5 \times \det((\mathbf{E}_i - \mathbf{E}_{i-1}) \circ (\mathbf{E}_{i+1} - \mathbf{E}_{i-1}))$
 - 11: **end for**
 - 12: **while** $\text{sizeof}(Q) > 2$ **do**
 - 13: $r \leftarrow \text{argmin}\{\text{DeltaH}\}$
 - 14: $Q \leftarrow Q \setminus \{Q_r\}$
 - 15: Update($\text{DeltaH}_{r-1}, \text{DeltaH}_{r+1}$)
 - 16: **end while**
 - 17: **end if**
 - 18: Return (DeltaH)
-

Algorithm 3 *Reduce* (Q, N)

Require: $Q \neq \emptyset$

- 1: Q is a solution set
- 2: N is the number of solutions will be discarded

Ensure: Reduce

- 3: $F = \text{empty}$
 - 4: Split Q into two subpopulation U and R // R is the collection of redundant individuals
 - 5: **if** $\text{sizeof}(R) \geq N$ **then**
 - 6: $F \leftarrow$ Random select N solutions from R
 - 7: $Q \leftarrow U \cup R \setminus F$
 - 8: **else**
 - 9: $F \leftarrow R$
 - 10: $\mathfrak{R}_1, \dots, \mathfrak{R}_v \leftarrow \text{Convexhull-based-sort-without-redundancy}(Q)$
 - 11: **for** $i = v \dots 1$ **do**
 - 12: **if** $\text{sizeof}(F) + \text{sizeof}(\mathfrak{R}_i) < N$ **then**
 - 13: $F \leftarrow F \cup \mathfrak{R}_i$
 - 14: $U = U \setminus \mathfrak{R}_i$
 - 15: **else**
 - 16: break
 - 17: **end if**
 - 18: **end for**
 - 19: $T \leftarrow \text{Select } (N - \text{sizeof}(F)) \text{ solutions with minial } \text{DeltaArea}(\mathfrak{R}_i)$
 - 20: $F \leftarrow F \cup T$
 - 21: $U \leftarrow U \setminus T$
 - 22: $Q \leftarrow U$
 - 23: **end if**
 - 24: Return (Q)
-

3.2. Area-based Selection Scheme

$$\Delta_{area} = \frac{\det((\mathbf{X} - \mathbf{L}) \circ (\mathbf{U} - \mathbf{X}))}{2} \quad (2)$$

In this subsection, we describe our area-based indicator for selection scheme in the new EMOA. The reason for why area-based and not hypervolume-based contribution is adopt is we need to maximize the area under the convex hull. Area-based indicator is more directly and efficiently. In the third graph of Fig. 4, it shows the novel area calculation for two dimensions. The contribution of one point x with its performance vector \mathbf{X} to the area is the area of triangle constructed by the point with its predecessor l and successor u with performance vector \mathbf{L} and \mathbf{U} . Alg. 2 gives the procedure of calculating of the novel area contribution. Eq. 2 gives the equation to how to calculate the area contribution of each point to its convex hull front.

Algorithm 4 CH-MOGP (Max, N)

Require: $Max > 0, N > 0$

- 1: Max is the maximum of evaluations
- 2: N is the population size

Ensure: CH-MOGP

- 3: $P_0 = init()$
 - 4: $t = 0$
 - 5: $m = 0$
 - 6: **while** $m < Max$ **do**
 - 7: $Q_t = empty$
 - 8: **for** $i = 1 : N$ **do**
 - 9: $q_i \leftarrow \text{Operators on } P_t$
 - 10: $Q_t \leftarrow Q_t + q_i$
 - 11: **end for**
 - 12: $P_{t+1} \leftarrow Reduce(P_t \cup Q_t)$
 - 13: $t \leftarrow t + 1$
 - 14: $m \leftarrow m + N$
 - 15: **end while**
-

3.3. CH-MOGP

Alg. 4 describes the CH-MOGP algorithm. The framework is very similar with SMS-EMOA and NSGA-II. However, we employ convex hull-based sorting without redundancy approach to rank the individuals into different levels. $(\mu + \mu)$ scheme is adopted into CH-MOGP to maximize the ROC performance. Because the target is to maximize area under the convex hull, area-based selection is designed instead of hypervolume-based contribution to keep the survivors with high area-based contribution.

In Alg. 4, first of all, the population size and the maximum of evaluations are given. Initial population is constructed by a group of solutions represented by genetic decision trees [29] using ramped-half-and-half method [30]. To generate the offsprings, two operators are employed and described in detail in [31]. The selection part of CH-MOGP

are operated by two schemes like other EMOAs, one is how to sort the population into different levels and the other is how to rank the solutions at the same level. Convex hull-based without redundancy sorting and area-based selection scheme play the main role to the selection part of CH-MOGP. To reduce the time of calling sorting approach, we also take $(\mu + \mu)$ scheme not $(\mu + 1)$ in SMS-EMOA.

4. Experimental Studies

4.1. Data Set

Nineteen data sets are selected from the UCI repository [32] and described in Table 2. Actually, we choose another three large-scaled data sets described in the last row of Table 2 to make more solid results. In this paper, we focus on binary classification problems, so all the data sets are two-class problems. Balanced and imbalanced benchmark data sets are carefully selected. The scale in terms of the number of instances of these data sets ranges from hundreds to thousands.

Table 1: Algorithms Involved

Name	Sorting	Selection	Scheme
<i>CH-MOGP</i>	CH-No-Redundancy	Area	$\mu + \mu$
<i>RCHH-EMOA</i>	CH-No-Redundancy	Area	$\mu + 1$
<i>CH-EMOA</i>	Convex Hull	Hypervolume	$\mu + 1$
<i>CHCrowding</i>	CH-No-Redundancy	Crowding-distance	$\mu + \mu$
<i>CHH-MOGP</i>	Convex Hull	Area	$\mu + 1$
<i>NSGA-II</i>	Non-dominated	Crowding-distance	$\mu + \mu$
<i>SMS-EMOA</i>	Non-dominated	Hypervolume	$\mu + 1$
<i>MOEA/D</i>	Fitness	Fitness	-

4.2. Algorithms Involved

To evaluate the performance of two strategies proposed in this paper, Table. 1 describes the algorithms involved to make rigorous and sufficient experimental comparisons. Generally speaking, this experiment is designed by considering three section of the EMOA, the first one is the strategy in sorting part including convex hull-based with and without redundancy sorting and non-dominated sorting(however, MOEA/D is decomposition based MOEA with different framework), the second one is the indicator for selection schemes including area-based, hypervolume-based and crowding-distance-based, the last one is related with $(\mu + \mu)$ and $(\mu + 1)$ for different EMOAs.

Table 2: Nineteen UCI Data Sets

Data Set	No. of features	Class Distribution	Data Set	No. of features	Class Distribution	Data Set	No. of features	Class Distribution
<i>australian</i>	14	383:307	<i>house-votes</i>	16	168:267	<i>pima</i>	8	268:500
<i>bcw</i>	9	458:241	<i>ionosphere</i>	34	225:126	<i>sonar</i>	60	97:111
<i>crx</i>	15	307:383	<i>kr-vs-kp</i>	36	1669:1527	<i>monks-3</i>	6	228:204
<i>transfusion</i>	4	178:570	<i>mammographic</i>	5	445:516	<i>spect</i>	22	212:55
<i>german</i>	24	700:300	<i>monks-1</i>	6	216:216	<i>parkinsons</i>	22	147:48
<i>wdbc</i>	30	212:357	<i>monks-2</i>	6	290:142	<i>tic-tac-toe</i>	9	626:332
<i>bands</i>	36	228:312						

Table 3: Evaluation Times for each algorithm on 19 UCI Data Sets

Data Set	No. of Evaluations	Data Set	No. of Evaluations	Data Set	No. of Evaluations	Data Set	No. of Evaluations
<i>australian</i>	100000	<i>bands</i>	150000	<i>bcw</i>	50000	<i>crx</i>	50000
<i>german</i>	200000	<i>house-votes</i>	30000	<i>ionosphere</i>	80000	<i>kr-vs-kp</i>	200000
<i>mammographic</i>	60000	<i>monks-1</i>	200000	<i>monks-2</i>	1000000	<i>monks-3</i>	40000
<i>parkinsons</i>	30000	<i>pima</i>	80000	<i>sonar</i>	30000	<i>spect</i>	40000
<i>tic-tac-toe</i>	300000	<i>transfusion</i>	22000	<i>wdbc</i>	30000	<i>adult</i>	10000
<i>magic04</i>	10000	<i>skin</i>	10000				

Table 4: Parameters for 8 algorithms

	Objective	Maximize Convex hull in ROC
Terminals of GP	{0,1} with 1 representing "Positive"; 0 representing "Negative"	Function set of GP If-then-else, and, or, not, >, <, =.
Data sets	22 UCI data sets	Algorithms 8 algorithms in Table 1
Crossover rate	0.9	Mutation rate 0.1
Shifting rate	0.1	Splitting rate 0.1
Parameters for GP	P(Population size) = 20; G (Maximum Evaluation Times) = M Number of Runs : 5 fold cross-validation 20 times	Termination criterion Maximum of G of evaluation time has been reached
Selection strategy	Tournament selection, Size = 4	Max depth of initial/inprocess individual program 3/17

4.3. Evaluation and Configuration

Evaluation: To evaluate the generalization performances of different classifiers produced by different algorithms, cross-validation is employed. We apply each algorithm on each 22 data sets with five-folds cross-validation for 20 times. Because we want to emphasize that our CH-MOGP could be better with less evaluation times, so we run each compared algorithms with large enough evaluation times to make them converge. Table. 3 gives the details for algorithms on each data set.

Configuration: We take the representation called GDT [29] as the individual in all multi-objective evolutionary algorithms. For binary classification problems, 0 and 1 (standing for negative and positive) are selected as the terminals of GP. Every classifier (individual) is constructed as *if-then-else* tree which involves *and*, *or*, *not*, *>*, *<* and *=* as operator symbols. Most offspring individuals are obtained by the crossover operator with probability 0.9. We also employed the shifting, and splitting operators described in [33] with probability 0.1. Tournament selection is adopted as the selection strategy and the tournament size is set to 4. To avoid overfitting, the maximum depth of each individual tree is limited to 17.

4.4. Results and Analysis

Fig. 7, Fig. 6, Table. 5 and Table. 11 show the performance of CH-EMOA compared with other EMOAs in 22 data sets. Generally speaking, CH-EMOA outperforms better not only at the AUCH but also the cost time.

In this subsection, we want to answer the questions as follows:

1. Why convex hull-based sorting without redundancy is better than traditional convex hull-based sorting?

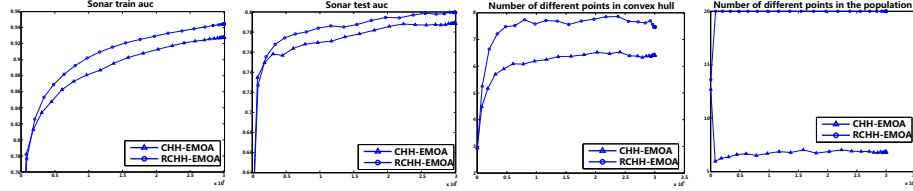


Figure 5: The diversity in convex hull-based sorting with and without redundancy effects the performance of the results

2. Is convex hull-based sorting without redundancy is better than non-dominated sorting approach in ROCCH maximization problem?
3. Is area-based selection scheme is comparable with or better than crowding-distance or hypervolume based selection?
4. Is CH-MOGP is better than NSGA-II, SMS-EMOA and MOEA/D for ROCCH maximization?
5. Does CH-MOGP show some advantages to traditional machine learning algorithms?

To evaluate the ideas we have proposed, we use 19 data sets in Table. 2 with algorithms described in Table. 1.

Table 5: Performance of four different frameworks of MOGP on UCI data sets, mean and standard deviation, multiplied by 100, are given in this table

	CH-MOGP	SMS-EMOA	NSGA-II	MOEA/D		CH-MOGP	SMS-EMOA	NSGA-II	MOEA/D
<i>australian</i>	91.49 ± 2.72	91.67 ± 2.48	91.16 ± 2.41	90.29 ± 2.75	<i>bands</i>	77.00 ± 4.05	76.38 ± 4.09	75.54 ± 3.56	71.85 ± 3.82
<i>bcw</i>	97.94 ± 1.20	97.73 ± 1.56	97.84 ± 1.41	97.48 ± 1.48	<i>crx</i>	91.30 ± 2.45	91.16 ± 2.33	91.14 ± 2.36	89.88 ± 2.51
<i>german</i>	73.10 ± 3.24	73.32 ± 3.33	72.39 ± 3.07	71.45 ± 2.85	<i>house-votes</i>	97.94 ± 1.56	97.69 ± 1.59	97.74 ± 1.71	97.15 ± 1.75
<i>ionosphere</i>	91.07 ± 4.95	90.51 ± 4.52	90.45 ± 4.53	89.89 ± 4.83	<i>kr-vs-kp</i>	98.40 ± 0.89	98.63 ± 0.75	98.39 ± 0.79	96.67 ± 1.43
<i>mammographic</i>	89.75 ± 2.01	89.48 ± 1.94	89.41 ± 1.87	87.50 ± 2.23	<i>monks-1</i>	99.70 ± 1.68	97.62 ± 3.71	99.62 ± 1.35	96.51 ± 5.69
<i>monks-2</i>	91.05 ± 8.00	89.28 ± 5.58	90.53 ± 5.19	73.26 ± 9.14	<i>monks-3</i>	99.81 ± 0.43	99.74 ± 0.45	99.45 ± 2.87	99.07 ± 0.88
<i>parkinsons</i>	86.79 ± 6.86	85.11 ± 6.68	84.90 ± 7.54	83.94 ± 6.72	<i>pima</i>	80.08 ± 3.38	79.85 ± 3.38	79.29 ± 3.70	76.93 ± 3.10
<i>sonar</i>	79.42 ± 5.87	78.04 ± 5.91	77.79 ± 7.34	75.75 ± 5.66	<i>spect</i>	77.38 ± 7.36	76.27 ± 7.14	76.91 ± 8.46	74.88 ± 6.43
<i>tic-tac-toe</i>	83.40 ± 10.4	79.56 ± 11.1	79.07 ± 13.4	70.85 ± 10.4	<i>transfusion</i>	71.62 ± 4.62	71.48 ± 4.47	71.49 ± 4.84	68.77 ± 4.63
<i>wdbc</i>	96.78 ± 1.92	96.49 ± 2.25	96.70 ± 2.11	95.90 ± 2.19					

Table 6: Performance of four different frameworks of MOGP on three big data sets, mean and standard deviation, multiplied by 100, are given in this table

	CH-MOGP	SMS-EMOA	NSGA-II	MOEA/D		CH-MOGP	SMS-EMOA	NSGA-II	MOEA/D
<i>adult</i>	84.58 ± 1.40	82.53 ± 2.15	84.01 ± 1.38	77.04 ± 2.54	<i>magic04</i>	83.02 ± 1.04	81.76 ± 1.57	82.01 ± 1.19	76.39 ± 3.07
<i>skin</i>	97.10 ± 1.11	95.46 ± 1.85	96.57 ± 1.25	93.20 ± 2.37					

4.4.1. Question 1

As we argued above, because of the greedy sort of convex hull-based sorting, the diversity will decrease fast as the generation or evaluation times. Fig. 5 shows the performance of CHH-EMOA and RCHH-EMOA which has been described in Table. 1. The only different between these two algorithms is the sorting scheme, CHH-EMOA adopts traditional convex hull-based sorting and RCHH-EMOA employs the convex hull-based sorting without redundancy approach. The third and fourth graph in Fig. 5 give the number of different individuals in the convex hull and in the whole population which are simply indicated as the measurement for the diversity. Obviously, RCHH-EMOA with larger diversity performs better than CHH-EMOA is the first and second graph in Fig. 5 which describe the AUCH performance in training and test data set (Here, we take data set "Sonar" as an example). However, we also give the Wilcoxon-Sum-Rank-Test results (Which is with a condence level of 0.95) of RCHH-EMOA and CHH-EMOA for 19 data sets in Table. 7. Generally speaking, RCHH-EMOA with convex hull-based sorting without redundancy is better than CHH-EMOA.

Table 7: Wilcoxon SUM-RANK Test on 19 UCI Data Sets: The table shows the wilcoxon sum test results between RCHH-EMOA and CHH-EMOA on 19 UCI Data sets at different evaluation times. Each $x-y-z$ in following table means RCH-EMOA wins x times, losses z times and draws y times. Ratio means the ratio of total evaluation times

Ratio	$\frac{1}{15}$	$\frac{1}{10}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	1
CHH-EMOA	4-15-0	5-14-0	5-14-0	6-13-0	6-13-0	6-13-0	4-15-0

4.4.2. Question 2

Algorithm CHCrowding and NSGA-II are involved into answering question 2. As described in Table. 1, CHCrowding and NSGA-II employ crowding-distance as the strategy into selection scheme, however, they adopt different sorting approach. Convex hull-based sorting without redundancy is employed into CHCrowding and NSGA-II takes fast nondominated sorting, which is the only difference between them. Table. 8 shows the Wilcoxon-Sum-Rank-Test results (Which is with a condence level of 0.95) for them. Obviously, CHCrowding losses none to NSGA-II and wins sometimes.

Table 8: Wilcoxon SUM-RANK Test on 19 UCI Data Sets: The table shows the wilcoxon sum test results between CHCrowding and NSGA-II on 19 UCI Data sets at different evaluation times. Each $x-y-z$ in following table means CHCrowding wins x times, losses z times and draws y times. Ratio means the ratio of total evaluation times

Ratio	$\frac{1}{15}$	$\frac{1}{10}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	1
NSGA-II	3-16-0	2-17-0	2-17-0	3-16-0	3-16-0	3-16-0	1-18-0

4.4.3. Question 3

For question 3, we takes two comparisons to explain. The first is that CHH-EMOA and CH-EMOA which are the same except for the selection schemes. In other words,

CHH-EMOA prefers area-based selection and hypervolume contribution is involved into selection scheme for CH-EMOA. We also gives the Wilcoxon-Sum-Rank-Test results (Which is with a condence level of 0.95) for them in Table. 9. Obviously, area-based selection works better than hypervolume contribution when they are combined with convex hull-based sorting approach into multi-objective optimization algorithm designs. On the other hand, we employ CHCrowding and CH-MOGP to measure the different performance of area-based selection and crowding-distance selection. However, Table. 10 shows there is no difference between them in 19 data sets. One reason is that convex hull-based sorting without redundancy plays more important role in the multi-objective algorithms than the selection scheme, however, selection scheme is also needed for the EMOAs. Though area-based and crowding-distance based selection schemes show no difference in above two algorithms, we still choose area-based selection because it is more intuitive for maximizing the ROC performance.

Table 9: Wilcoxon SUM-RANK Test on 19 UCI Data Sets: The table shows the wilcoxon sum test results between CHH-EMOA and CH-EMOA on 19 UCI Data sets at different evaluation times. Each $x-y-z$ in following table means CHH-EMOA wins x times, losses z times and draws y times. Ratio means the ratio of total evaluation times

Ratio of total evaluations	$\frac{1}{15}$	$\frac{1}{10}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	1
CH-EMOA	3-16-0	4-15-0	4-15-0	4-15-0	4-15-0	6-13-0	5-14-0

Table 10: Wilcoxon SUM Test on 19 UCI Data Sets: The table shows the wilcoxon sum test results between CH-MOGP and CHCrowding on 19 UCI Data sets at different evaluation times. Each $x-y-z$ in following table means CH-MOGP wins x times, losses z times and draws y times. Ratio means the ratio of total evaluation times

Ratio	$\frac{1}{15}$	$\frac{1}{10}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	1
CHCrowding	0-19-0	0-19-0	0-19-0	0-19-0	0-19-0	0-19-0	0-19-0

4.4.4. Question 4

AUCH analysis: To answer the question 4, we employ more data set, specially for big data set because we always emphasize that our algorithm will perform better with less evaluation times which means we will save a lot of time for problems with expensive evaluation. Table. 12 describes three big data set. Table. 5 and 6 give the result of 4 different evolutionary multi-objective algorithms involved with GDT for maximizing the area under convex hull in ROC space. Furthermore, Table. 11 gives the Wilcoxon Sum-Rank Test results (Which is with a condence level of 0.95) for them. To compare the performance of all algorithms at each stage of its evolutionary process, we show the results at $1/15$, $1/10$, $1/4$, $1/3$, $1/2$ and 1 of the whole process. It is very clear that CH-MOGP outperforms among these EMOAs.

The Performance and Evaluation Times: Fig. 7 and Fig. 6 show the performance of CH-MOGP, SMS-EMOA, NSGA-II and MOEA/D on 22 data sets. Actually, we

Table 11: Wilcoxon SUM-Rank Test on 22 UCI Data Sets: The table shows the wilcoxon sum test results between CH-EMOA and other three EMOAs (NSGA-II, SMS-EMOA and MOEA/D) on 22 UCI Data sets at different evaluation times. Each $x-y-z$ in following table means CH-EMOA wins x times, losses z times and draws y times. Ratio means the ratio of total evaluation times

Ratio	$\frac{1}{15}$	$\frac{1}{10}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	1
<i>NSGA-II</i>	4-15-0	4-15-0	2-17-0	4-15-0	5-14-0	5-14-0	4-15-0
<i>SMS-EMOA</i>	11-8-0	11-8-0	6-13-0	5-14-0	4-15-0	4-15-0	5-14-0
<i>MOEA/D</i>	19-0-0	19-0-0	19-0-0	19-0-0	19-0-0	19-0-0	19-0-0
<i>NSGA-II</i>	0-3-0	1-2-0	1-2-0	1-2-0	2-1-0	2-1-0	2-1-0
<i>SMS-EMOA</i>	3-0-0	3-0-0	3-0-0	3-0-0	3-0-0	3-0-0	3-0-0
<i>MOEA/D</i>	3-0-0	3-0-0	3-0-0	3-0-0	3-0-0	3-0-0	3-0-0

Table 12: Three Large-scaled Data Sets

Data Set	No. of features	Class Distribution	Data Set	No. of features	Class Distribution	Data Set	No. of features	Class Distribution
<i>skin</i>	4	50859:194198	<i>magic04</i>	10	12332 :6688	<i>adult</i>	14	11687 : 37155

Table 13: Evaluation Times for each algorithm on 22 UCI Data Sets

Data Set	No. of Evaluations	Data Set	No. of Evaluations	Data Set	No. of Evaluations	Data Set	No. of Evaluations
<i>australian</i>	100000	<i>bands</i>	1500000	<i>bcw</i>	18500	<i>crx</i>	450000
<i>german</i>	120000	<i>house-votes</i>	24000	<i>ionosphere</i>	80000	<i>kr-vs-kp</i>	2000000
<i>mammographic</i>	80000	<i>monks-1</i>	230000	<i>monks-2</i>	10000000	<i>monks-3</i>	190000
<i>parkinsons</i>	42000	<i>pima</i>	180000	<i>sonar</i>	12000	<i>spect</i>	10000
<i>tic-tac-toe</i>	3000000	<i>transfusion</i>	35000	<i>wdbc</i>	21000	<i>adult</i>	300000
<i>magic04</i>	40000	<i>skin</i>	30000				

Table 14: Performance of CH-MOGP and traditional classifiers on UCI data sets, mean and standard deviation, multiplied by 100, are given in this table

	CH-MOGP	C4.5	NB	Pyrirel		CH-MOGP	C4.5	NB	Pyrirel
<i>australian</i>	91.97 \pm 2.53	85.52 \pm 4.05	89.47 \pm 2.78	91.75 \pm 2.36	<i>monks-3</i>	100.0 \pm 0.00	100.0 \pm 0.00	95.94 \pm 2.17	99.60 \pm 0.27
<i>bands</i>	78.50 \pm 3.56	74.56 \pm 4.59	73.91 \pm 4.68	76.07 \pm 4.81	<i>parkinsons</i>	86.10 \pm 6.66	78.91 \pm 9.76	85.91 \pm 6.11	88.24 \pm 5.83
<i>bcw</i>	98.17 \pm 1.06	95.05 \pm 2.55	98.92 \pm 0.62	98.16 \pm 1.09	<i>pima</i>	80.74 \pm 3.12	75.23 \pm 4.93	81.40 \pm 3.01	79.58 \pm 2.92
<i>crx</i>	91.82 \pm 2.27	85.51 \pm 3.94	87.88 \pm 3.16	90.65 \pm 2.77	<i>sonar</i>	81.44 \pm 5.15	73.85 \pm 7.84	80.12 \pm 7.03	69.92 \pm 8.64
<i>german</i>	74.27 \pm 2.79	65.36 \pm 4.74	78.42 \pm 2.94	75.95 \pm 3.25	<i>spect</i>	78.56 \pm 7.44	76.88 \pm 8.91	84.09 \pm 6.03	83.51 \pm 7.01
<i>house-votes</i>	98.23 \pm 1.26	96.35 \pm 2.04	98.05 \pm 1.04	97.80 \pm 1.49	<i>tic-tac-toe</i>	90.07 \pm 8.88	84.91 \pm 13.9	61.50 \pm 14.7	70.41 \pm 12.5
<i>ionosphere</i>	92.42 \pm 3.66	88.20 \pm 5.65	93.57 \pm 3.18	93.68 \pm 4.23	<i>transfusion</i>	72.19 \pm 4.89	71.08 \pm 5.08	70.93 \pm 4.94	70.87 \pm 5.39
<i>kr-vs-kp</i>	99.40 \pm 0.26	99.71 \pm 0.23	93.21 \pm 1.00	98.26 \pm 0.44	<i>wdbc</i>	97.32 \pm 1.40	92.74 \pm 3.16	98.14 \pm 1.33	96.58 \pm 1.94
<i>mammographic</i>	90.20 \pm 1.76	87.66 \pm 2.21	89.77 \pm 1.96	89.70 \pm 2.02	<i>adult</i>	88.97 \pm 0.37	88.89 \pm 0.53	85.27 \pm 0.37	90.37 \pm 0.25
<i>monks-1</i>	100.0 \pm 0.00	77.13 \pm 6.90	73.18 \pm 4.58	70.93 \pm 5.59	<i>magic04</i>	87.16 \pm 0.74	86.76 \pm 0.83	75.70 \pm 0.74	85.37 \pm 0.76
<i>monks-2</i>	95.68 \pm 4.61	94.17 \pm 5.93	52.38 \pm 7.04	51.25 \pm 6.16	<i>skin</i>	99.49 \pm 0.11	99.93 \pm 0.02	94.17 \pm 0.07	98.15 \pm 0.08

Table 15: Wilcoxon SUM Test on 22 UCI Data Sets: The table shows the wilcoxon SUM-RANK test results between CH-EMOA and other three machine learning algorithms on 22 UCI Data sets at different evaluation times. Each $x-y-z$ in following table means CH-EMOA wins x times, losses z times and draws y times.

	CH-MOGP	C4.5	NB	PRIE
<i>CH-MOGP</i>		15-5-2	11-6-5	13-4-5
<i>C4.5</i>			8-2-12	8-1-13
<i>NB</i>				6-6-10
<i>Pyrirel</i>				

give the convergence of these EMOAs for training and test data sets with 5-fold cross-validation 20 times. Generally speaking, the curves of CH-MOGP are over others on most data sets. In other words, for a given and very limited evaluation times, CH-MOGP can perform better than other EMOAs in the classification task.

4.4.5. Question 5

AUCH comparison: In this sub-section, we compare CH-MOGP with C4.5 [34], Naive Bayes(NB) [35] and PRIE [7] which are traditional machine learning algorithms for constructing classifiers. To make a fair comparison, we set the population size of CH-MOGP as 100. The reason is that soft classifiers usually output scores/probabilities to its test data sets, and the number of different kinds of scores or probabilities decides the number of performance points in ROC space, however, that number is not a small one. So we choose a general number, 100, as the population size of CH-MOGP. Meanwhile, it needs more evaluations to a larger population size, so Table. 13 gives the evaluation times for CH-MOGP in 22 data sets. Fig. 14 shows the results for CH-MOGP, C4.5, NB and PRIE in all data sets, furthermore, Wilcoxon Sum-Rank Test results (Which is with a condence level of 0.95) are given in Fig. 15.

Evaluation Times:

Table 16: Times for CH-MOGP, C4.5, NB and PRIE to construct classifiers to maximize ROCCH

Time(s)	CHMOGP	C4.5	NB	PRIE	Time(s)	CHMOGP	C4.5	NB	PRIE
australian	116.91	0.06	0.02	4.18	bands	2242.5	0.04	0.03	15.85
bcw	28.63	0.01	0.02	0.53	crx	653.45	0.02	0.02	2.92
german	234.27	0.16	0.04	4.79	house-votes	13.2	0.01	0.02	0.48
ionosphere	59.51	0.04	0.02	5.77	kr-vs-kp	12389.37	0.27	0.22	1.58
mammographic	95.75	0.01	0.02	0.87	monks-1	174.67	0.01	0.02	0.29
monks-2	8558.14	0.01	0.02	0.3	monks-3	83.49	0.01	0.02	0.31
parkinsons	17.48	0.01	0.02	1.62	pima	206.04	0.02	0.02	16.46
sonar	129.28	0.03	0.02	31.45	spect	89.05	0.02	0.02	0.39
tic-tac-toe	5396.3	0.03	0.02	0.48	transfusion	28.98	0.01	0.02	4.34
wdbc	27.39	0.04	0.03	20.86	adult	15655.92	0.42	2.08	1771.73
magic04	7601.82	0.28	0.57	1103.05	skin	91856.38	15.01	3.7	70.15

Table. 16 gives the cost time for CH-MOGP, C4.5, NB and PRIE to construct classifiers to maximize ROCCH. The experiment environment is an 8 core CPU with 2.13GHz and 24GB RAM. Obviously, CH-MOGP consumes much more time than others, because of the metaheuristic character of EAs, GP needs to evaluate many classiers until it converges. On the other hand, NB method calculates an a posteriori probability and the C4.5 adopts uses a greedy method to increase information gain. PRIE employs a greedy strategy to construct classiers (more than one, usually dozens of classiers) to maximize the ROCCH, so it cost a little more time than NB and C4.5, but still much less than CH-MOGP. Actually, how to reduce the evaluation time of CH-MOGP is an important topic.

5. Conclusions and Future Work

In this paper, we propose convex hull-based sorting approach and area-based selection scheme involved into multi-objective genetic programming for maximizing the

ROC performance in classification tasks. First, we emphasized that convex hull maximization problems is similar but beyond multi-objective optimization problem, traditional techniques are helpful but needed to improve the solve the this kinds of problem. Instead of fast non-dominated sorting approach in NSGA-II and SMS-EMOA, convex hull-based sorting is investigated in new algorithm design, however, we found convex hull-based sorting without redundancy was efficient to avoid losing diversity in the search process. Area-based selection scheme with $\mu + \mu$ is also designed for helping to rank the population. The new algorithm- CH-MOGP is also performed on benchmarks and work better than other traditional EMOAs and some other traditional machine learning algorithms. In the future work, there are three topics would be discussed. The first is how to improve CH-MOGP to reduce its time consuming character but keep the comparable performance for ROCCH maximization. The second one is that GP-based classifier could be replaced by other tree-based classifiers or other traditional machine learning classifiers such as SVM, NB, etc.. Different classifier would result better performance for ROCCH maximization. The third topic is convex hull based without redundancy sorting and area-based selection scheme, these two strategies are not only used in classification but also other area such as numerical optimization.

Acknowledgment

The authors would like to thank...

References

- [1] T. Fawcett, An introduction to roc analysis, *Pattern recognition letters* 27 (8) (2006) 861–874.
- [2] J. Egan, *Signal detection theory and roc analysis*.
- [3] H. Sox, M. Higgins, *Medical decision making*, Amer College of Physicians, 1988.
- [4] J. Swets, Measuring the accuracy of diagnostic systems., *Science* 240(4857) (1988) 1285–1293.
- [5] F. Provost, T. Fawcett, Robust classification for imprecise environments, *Machine Learning* 42 (3) (2001) 203–231.
- [6] T. Fawcett, Using rule sets to maximize roc performance, in: *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, IEEE, 2001, pp. 131–138.
- [7] T. Fawcett, Prie: a system for generating rulelists to maximize roc performance, *Data Mining and Knowledge Discovery* 17 (2) (2008) 207–224.
- [8] P. Flach, S. Wu, Repairing concavities in roc curves, in: *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI05)*, 2005, pp. 702–707.

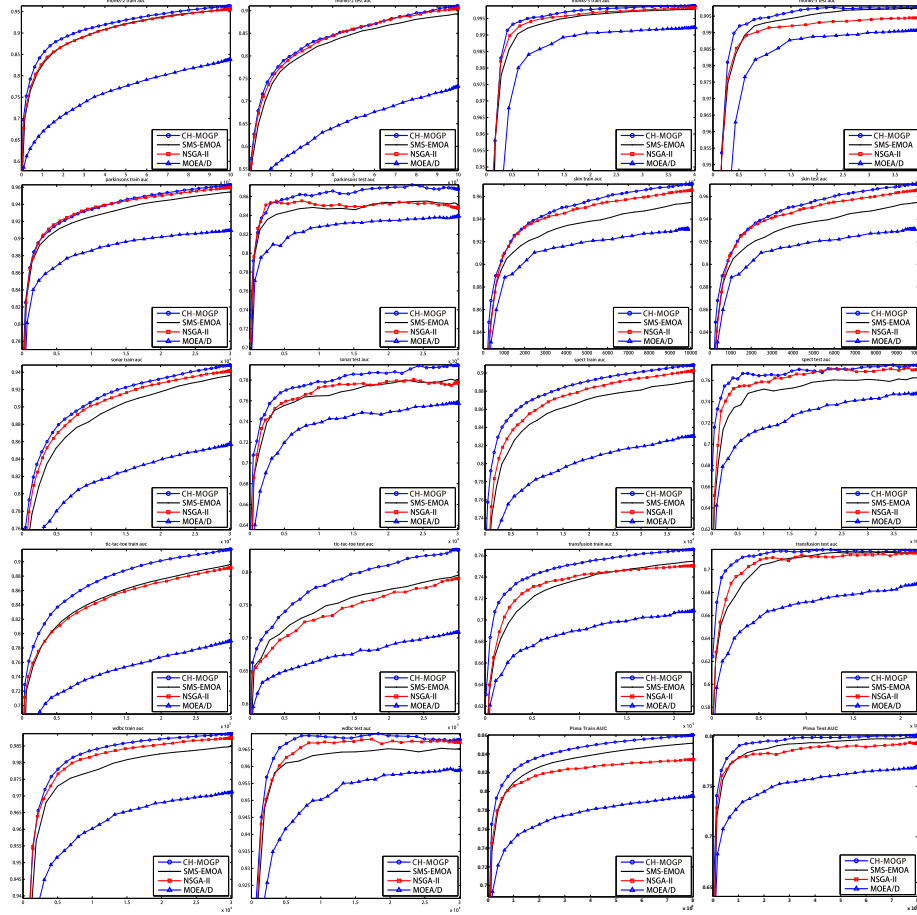


Figure 6: Performance of four different EMOAs on 10 data sets

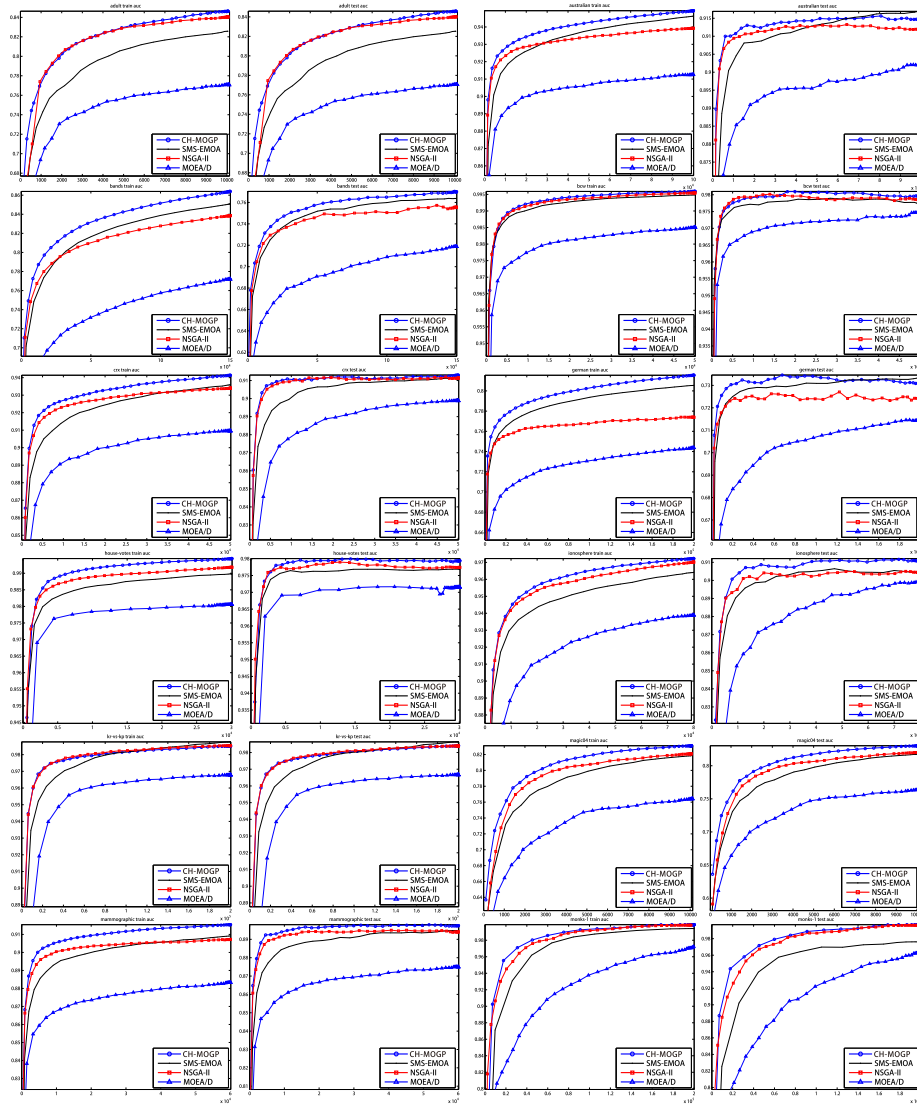


Figure 7: Performance of four different EMOAs on 12 data sets

- [9] R. Prati, P. Flach, Roccer: an algorithm for rule learning based on roc analysis, in: Proceedings of the 19th international joint conference on Artificial intelligence, Morgan Kaufmann Publishers Inc., 2005, pp. 823–828.
- [10] M. Barreno, A. A. Cárdenas, J. D. Tygar, Optimal roc curve for a combination of classifiers, *Advances in Neural Information Processing Systems* 20 20 (X) (2008) 57–64.
- [11] H. Zhao, A multi-objective genetic programming approach to developing pareto optimal decision trees, *Decision Support Systems* 43 (3) (2007) 809–826.
- [12] U. Bhowan, M. Zhang, M. Johnston, Multi-objective genetic programming for classification with unbalanced data, *AI 2009: Advances in Artificial Intelligence* (2009) 370–380.
- [13] U. Bhowan, M. Johnston, M. Zhang, X. Yao, Evolving diverse ensembles using genetic programming for classification with unbalanced data.
- [14] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, *Evolutionary Computation, IEEE Transactions on* 6 (2) (2002) 182–197.
- [15] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, *Evolutionary Computation, IEEE Transactions on* 11 (6) (2007) 712–731.
- [16] N. Beume, B. Naujoks, M. Emmerich, Sms-emoa: Multiobjective selection based on dominated hypervolume, *European Journal of Operational Research* 181 (3) (2007) 1653 – 1669. doi:10.1016/j.ejor.2006.08.008", url={http://www.sciencedirect.com/science/article/pii/S0377221706005443}.
- [17] K. Bringmann, T. Friedrich, F. Neumann, M. Wagner, Approximation-guided evolutionary multi-objective optimization, *Proceeding of the Twenty-Second International Joint Conference on Artificial Intelligence (Barcelona, Spain, 2011)* 1198–1203.
- [18] P. Wang, K. Tang, T. Weise, E. Tsang, X. Yao, Multiobjective genetic programming for maximizing roc performance, *Neurocomputing*.
- [19] T. Fawcett, Roc graphs: Notes and practical considerations for researchers, *Machine Learning* 31 (2004) 1–38.
- [20] J. Shan-Fan, S. Xiong, J. Zhuo-Wang, The multi-objective differential evolution algorithm based on quick convex hull algorithms, in: *Natural Computation, 2009. ICNC'09. Fifth International Conference on*, Vol. 4, IEEE, 2009, pp. 469–473.
- [21] M. D. Monfared, A. Mohades, J. Rezaei, Convex hull ranking algorithm for multi-objective evolutionary algorithms, *Scientia Iranica* 18 (6) (2011) 1435 – 1442. doi:10.1016/j.scient.2011.08.017.

URL <http://www.sciencedirect.com/science/article/pii/S1026309811001672>

- [22] S. Zapotecas Martínez, C. A. Coello Coello, A novel diversification strategy for multi-objective evolutionary algorithms, in: Proceedings of the 12th annual conference companion on Genetic and evolutionary computation, GECCO '10, ACM, New York, NY, USA, 2010, pp. 2031–2034. doi:10.1145/1830761.1830852.
URL <http://doi.acm.org/10.1145/1830761.1830852>
- [23] M. Cococcioni, P. Ducange, B. Lazzerini, F. Marcelloni, A new multi-objective evolutionary algorithm based on convex hull for binary classifier optimization, in: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, 2007, pp. 3150–3156.
- [24] P. Ducange, B. Lazzerini, F. Marcelloni, Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets, Soft Computing-A Fusion of Foundations, Methodologies and Applications 14 (7) (2010) 713–728.
- [25] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evolutionary Computation 15 (1) (2007) 1–28.
- [26] T. Weise, Global Optimization Algorithms – Theory and Application, it-weise.de (self-published): Germany, 2009.
URL <http://www.it-weise.de/projects/book.pdf>
- [27] P. Flach, Roc analysis, Encyclopedia of Machine Learning. Berlin Heidelberg: Springer (2010) 869–874.
- [28] R. Jarvis, On the identification of the convex hull of a finite set of points in the plane, Information Processing Letters 2 (1) (1973) 18–21.
- [29] L. Jin, FGP: A genetic programming based financial forecasting tool, Phd Thesis, Citeseer, 2000.
- [30] R. Poli, W. Langdon, N. McPhee, A field guide to genetic programming, Lulu Enterprises Uk Ltd, 2008.
- [31] P. Wang, K. Tang, E. Tsang, X. Yao, A memetic genetic programming with decision tree-based local search for classification problems, in: Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE, 2011, pp. 917–924.
- [32] UCI, UCI machine learning repository (2009).
URL <http://archive.ics.uci.edu/ml>
- [33] P. Wang, E. P. Tsang, T. Weise, K. Tang, X. Yao, Using gp to evolve decision rules for classification in financial data sets, Special Session on Evolutionary Computing of the 9th IEEE International Conference on Cognitive Informatics (ICCI 2010) (2010) 722–727.

- [34] J. Quinlan, C4. 5: programs for machine learning, Morgan Kaufmann, 1993.
- [35] D. Lewis, Naive (bayes) at forty: The independence assumption in information retrieval, Machine Learning: ECML-98 (1998) 4–15.